

REMARKS

Applicant appreciates the time taken by the Examiner to review Applicant's present application. This application has been carefully reviewed in light of the Official Action mailed October 15, 2007 ("Final Office Action"). This Reply encompasses a bona fide attempt to address the rejections raised by the Examiner and presents amendments as well as reasons why Applicant believes that the claimed invention is novel and unobvious over the applied prior art. Accordingly, Applicant respectfully requests reconsideration and favorable action in this case.

Claim Status

Claims 49 and 51-60 were rejected. Claims 1-48, 50, and 61-64 were cancelled previously. Claims 65-73 are newly added. Support for the amendments to the claims can be found in the Specification as originally filed, particularly paragraphs 21-22, 30-31, 38, 42, 50-51, and 54-58. No new matter is introduced. By this Amendment, claims 49, 51-60, and 65-73 are pending.

Rejections under 35 U.S.C. § 103

Claims 49, 51-52, and 56-59 stand rejected under 35 U.S.C. § 103 as being unpatentable over U.S. Patent Application Publication No. 2003/0105770 ("MacLeod1") in view of U.S. Patent Application Publication No. 2003/0105654 ("MacLeod2"). Claims 53-55 and 60 stand rejected under 35 U.S.C. § 103 over MacLeod1 in view of MacLeod2 as applied to claims 49, 51-52, and 56-59 above and further in view of U.S. Patent Application Publication No. 2004/0187100 ("Thiruvillamalai"). Arguments submitted in the previous reply filed July 30, 2007 ("Previous Reply") remain pertinent and are therefore expressly incorporated herein by reference. Additionally, Applicant respectfully submits the following for the Examiner's consideration.

On page 13 of the Final Office Action, the Examiner insisted that MacLeod1 teaches the limitations of "analyzing a set of data and generating a set of content types to represent the set of data based on the analysis of the data" as MacLeod1 teaches that a content class models a set of items that have similar properties and fulfill similar purposes, citing paragraph 22 of

MacLeod1. Applicant respectfully disagrees. The cited portion of MacLeod1 explicitly describes:

A schema is a collection of content classes and associations that abstract items, or "objects" that tangibly or intangibly exist in the real world. A content class models a set of items that have similar properties and fulfill similar purposes. A content class defines the purpose or content of an item by containing as its elements a list of properties appropriate for that purpose or content. Content classes imply a set of semantic requirements for the item. Content classes follow a hierarchical structure.

Thus, by definition, the content classes of MacLeod1 are not content types as particularly described in at least paragraph 36 of the Specification. As an example, in MacLeod1, a programmer or application designer creates new structural object classes or attributes based on the flexible attribute to support new application data requirements, versions, and so on. See MacLeod1, paragraph 58. By contrast, in embodiments as claimed, a business user, as defined in the Specification, paragraph 25, can generate, modify, or delete content types without having to know a programming language as the content types essentially replace conventional templates that required programmers to code. See Specification, paragraphs 36-37. Note that the omission of an element and retention of its function is an indicia of unobviousness. *In re Edge*, 359 F.2d 896, 149 USPQ 556 (CCPA 1966) (emphasis in original).

As submitted in the Previous Reply, MacLeod1 teaches a directory schema with flexible attribute content class 418 that can be extended with new structural object classes or attributes based there on without needing to modify the directory schema. See MacLeod1, paragraphs 57-58. MacLeod1 explicitly describes that a directory schema is a collection of base content classes and associations and that directory schemas are typically very carefully designed (and hence not to be readily altered) to provide content classes to meet present and future requirements of a directory. See MacLeod1, paragraphs 2-3.

What MacLeod1 lacks, and the Final Office Action seems to acknowledge, is, in general, any description on how to integrate legacy data into a content management system and, in particular, the limitation of "wherein one of the content types comprises a policy annotation, the policy annotation comprising management information including a workflow corresponding to the content type." In the Final Office Action, page 6, the Examiner applies

claims 72-73 of MacLeod2 to this particular limitation. However, claim 72 of MacLeod2 is directed to a computing device as recited in claim 55 of MacLeod2, which recites, among others, “mapping the state change to the object to a workflow comprising a set of tasks.” Claim 72 of MacLeod2 further recites “wherein the tasks implement a policy with respect to one or more directory resources.” The workflow recited in claim 72 of MacLeod2 seems to reference the workflow described in paragraphs 108-109 of MacLeod2. Thus, claim 72 of MacLeod2 appears to be directed to a computer device that has a processing means for mapping the state change to the object to a workflow which comprises a set of tasks that implement a policy with respect to one or more directory resources. That is, claim 72 of MacLeod2 does not describe that one of the content types comprises a policy annotation, which comprises management information, which includes a workflow, which corresponds to the content type.

Similarly, claims 73 and 74 of MacLeod2 recite a workflow enable directory schema comprising a plurality of base object content classes. Nowhere in claims 72-74 does MacLeod2 teach integrating data into a content management system in general or “wherein one of the content types comprises a policy annotation, the policy annotation comprising management information including a workflow corresponding to the content type” in particular.

At best, MacLeod1 and MacLeod2 seem to provide a directory schema with a flexible content object class that can be used for workflow management. Still, the cited portions of MacLeod1 and MacLeod2 explicitly lack any description of “a method for integrating data into a content management system,” “generating a set of content types,” “one of the content types comprises a policy annotation,” “the policy annotation comprising management information,” or “management information including a workflow corresponding to the content type,” as recited in claim 49.

In the Final Office Action, page 14, the Examiner further insisted that MacLeod1 teaches the limitation of “generating a set of content instance objects from the content type objects,” citing Figure 6 and paragraphs 25 and 74 of MacLeod1. Specifically, the “Examiner interprets object instance [*sic*] as a content type object of a data type of particular class.” Applicant respectfully disagrees. Such an interpretation presents several problems. First, it is inconsistent with the explicit definitions of terms provided in Applicant’s disclosure.

Where an explicit definition is provided by the applicant for a term, that definition will control interpretation of the term as it is used in the claim. *Toro Co. v. White Consolidated Industries Inc.*, 199

F.3d 1295, 1301, 53 USPQ2d 1065, 1069 (Fed. Cir. 1999)
(meaning of words used in a claim is not construed in a
"lexicographic vacuum, but in the context of the specification and
drawings").

For example, as explicitly defined in the Specification, a "content instance object" refers to an instantiation of a content instance. See Specification, paragraph 24. The Examiner's interpretation is inconsistent with this definition.

During patent examination, the pending claims must be given the
broadest reasonable interpretation consistent with the
specification. *In re Morris*, 127 F.3d 1048, 1054, 44 USPQ2d
1023, 1027 (Fed. Cir. 1997) (emphasis added); *In re Prater*, 415
F.2d 1393, 162 USPQ 541 (CCPA 1969). See also MPEP § 2111
- § 2111.01.

Second, the Examiner's interpretation appears to be inconsistent with the language of claim 49. Specifically, the method according to claim 49 includes three generating acts:

- first, generating a set of content types;
- second, generating a set of content type objects corresponding to the set of content types; and
- third, generating a set of content instance objects from the content type objects.

In the Final Office Action, the first act of "generating a set of content types" was not mentioned with respect to claim 49 on pages 3-6. However, in responding to Applicant's arguments submitted in the Previous Reply, the Examiner applied MacLeod1's teaching of a content class to this claim limitation. See Final Office Action, pages 13-14. The Examiner's argument with respect to the first act of "generating a set of content types" on pages 13-14 of the Final Office Action appears to be the same as the rejection applied to the second act of "generating a set of content type objects corresponding to the set of content types" on pages 3-4 of the Final Office Action. The Examiner therefore has either impermissibly omitted a claim limitation or improperly applied the same argument to two claim limitations. In either case, the Examiner's interpretation is inconsistent with the language of claim 49.

Third, the Examiner's interpretation appears to be inconsistent with the interpretation that those skilled in the art would reach in light of the specification.

The Patent and Trademark Office ("PTO") determines the scope of claims in patent applications not solely on the basis of the claim language, but upon giving claims their broadest reasonable construction "in light of the specification as it would be interpreted by one of ordinary skill in the art." *In re Am. Acad. of Sci. Tech. Ctr.*, 367 F.3d 1359, 1364[, 70 USPQ2d 1827] (Fed. Cir. 2004). The broadest reasonable interpretation of the claims must also be consistent with the interpretation that those skilled in the art would reach. *In re Cortright*, 165 F.3d 1353, 1359, 49 USPQ2d 1464, 1468 (Fed. Cir. 1999).

The Examiner interprets MacLeod1's "object instance" as a content type object of a data type of a particular class. However, in light of the Specification, particularly paragraphs 23-24, one of ordinary skill in the art, at the time the invention was made, would have reached an interpretation of claim 49 that is inconsistent with that of the Examiner's. Moreover, claim 49 particularly distinguishes three generating acts: generating a set of content types, generating a set of content type objects corresponding to the set of content types, and generating a set of content instance objects from the content type objects. By contrast, the Examiner's interpretation does not seem to recognize these three distinct acts. Note that the Specification also distinguishes "data types" and "content types". In the Specification, a "data type" is defined via an XML Schema Definition ("XSD") for a valid XML document in lieu of a Document Type Definition ("DTD") and a "content type" is generated so that it overrides a content management application's presentation for a Java Server Page ("JSP"). See Specification, paragraph 50.

On pages 14-15 of the Final Office Action, the Examiner insisted that MacLeod1 teaches the limitation of "associating each of the set of data with at least one of the content instance objects, wherein at least one of the content instance objects is associated with two or more datum of the set of data, each of the datum residing in a distinct data storage," citing Figures 5-6, paragraphs 75 and 77, and Abstract of MacLeod1. Applicant respectfully disagrees and notes that MacLeod1 does not describe this limitation at least because MacLeod1 does not describe that "each of the datum residing in a distinct data storage." MacLeod1 explicitly describes, in paragraph 72:

The directory service 516 stores information about objects such as computers and resources on a network and makes this information available to users and network administrators in a

directory that ties disparate databases 506, or “directories” of data together into a single, logical directory, or “metadirectory”.

Specifically, the directory service manages and maintains a distributed directory that is based on the directory schema 400 with flexible attributes.

Thus, MacLeod1 does not describe the databases 506 as distinct data storages. They might be disparate, but they are logically indistinct. If the databases 506 of MacLeod1 were distinct data storages, they would change the principle operation of MacLeod1 as they would have logically distinct directories. By contrast, embodiments as claimed in claim 49 are not directed to a single, logical directory or any particular directory schema. With embodiments as claimed in claim 49, users can model content types in terms of a business’s needs and in a business’s own vocabulary. See Specification, paragraph 38. That is, with embodiments as claimed in claim 49, content types can be generated to model an existing user schema and content instance objects of the relevant content types can be automatically created thereafter. See Specification, paragraph 52. By associating the data from distinct data storages with these content instance objects thus created as recited in claim 49, such data can be brought into a content management system and the content management system can take over management of the data. See Specification, paragraph 52.

Furthermore, Applicant respectfully submits that MacLeod1 does not describe such an “associating” act. As noted above, MacLeod1 describes that the directory service manages and maintains a distributed directory that is based on the directory schema 400. Accordingly, in MacLeod1, an application that has instantiated or that is using the first object instance already knows of the first object instance’s interface and how to unpack and use the first data string. Thus, MacLeod1 does not describe associating a content instance object with data residing in distinct data storages.

In view of the foregoing and arguments submitted in the Previous Reply, claim 49 is submitted to be patentable under 35 U.S.C. § 103 over MacLeod1 and MacLeod2 and therefore should be allowed. For similar reasons, dependent claims 50-60 are submitted to be patentable under 35 U.S.C. § 103 over MacLeod1 and MacLeod2. Accordingly, withdrawal of the rejections is respectfully requested.

Newly Added Claims 65-73 Recite Subject Matter Not Reached by Prior Art

Newly added claims 65-73 recite subject matter not reached by the applied prior art for similar reasons as submitted above with respect to claim 49. Specifically, independent claim 65 recites a method for integrating data into a content management system, comprising:

- enabling a user to define content types in terms of said user's business context;
- creating a content type object for each of said content types;
- locating data to be integrated into said content management system;
- creating a content instance object for each piece of data which matches a content type;
- associating or attaching said content instance object to said piece of data which said content instance object represents;
- persisting said content instance object in said content management system; and
- managing said piece of data using said content instance object.

Independent claim 70 recites similar limitations. Thus, claims 65-73 are directed to embodiments for integrating data into a content management system. Specifically, with embodiments as claimed in claims 65-73, a user can define content types in terms of the user's business context. See Specification, paragraphs 21-22. Such a user may be a person with little, if any, training in computer programming and writing code in a programming language. See Specification, paragraphs 25 and 36-38. As submitted above, the applied art depends on a programmer or application designer to create new structural object classes or attributes based on the flexible attribute to support new application data requirements, versions, and so on. *Supra*.

Once content type objects are created from the user-defined content types, data to be integrated into the content management system (e.g., legacy data residing in a database) which match a content type can be located. See Specification, paragraph 57. In some embodiments, this can be done automatically (e.g., using a Java program). *Id.* As submitted above, the applied art appears to be silent on integrating legacy data into a content management system.

For each piece of data which matches a content type, a content instance object is created and persisted in the content management system. *Id.* The content instance object can then be associated or attached to the particular piece of data which the content instance object represents. See Specification, paragraph 58. Using the saved content instance objects, the content management system can manage the legacy data without the need to manually enter the data or to modify the data in any way. *Id.* The applied art neither teaches nor suggests this.


CONCLUSION

Applicant has now made an earnest attempt to place this case in condition for allowance. Other than as explicitly set forth above, this reply does not include an acquiescence to statements, assertions, assumptions, conclusions, or any combination thereof in the Office Action. For the foregoing reasons and for other reasons clearly apparent, Applicant respectfully requests full allowance of Claims 49, 51-60, and 65-73. The Examiner is invited to telephone the undersigned at the number listed below for discussing an Examiner's Amendment or any suggested actions for accelerating prosecution and moving the present application to allowance.

A Request for Continued Examination is enclosed. The Director of the U.S. Patent and Trademark Office is hereby authorized to charge the RCE fee and additional fees owing and credit any overpayments to Deposit Account No. 50-3183 of Sprinkle IP Law Group.

Respectfully submitted,

Sprinkle IP Law Group
Attorneys for Applicant



Katharina W. Schuster
Reg. No. 50,000

Date: October 31, 2007

1301 W. 25th Street, Suite 408
Austin, TX 78705
Tel. (512) 637-9220
Fax. (512) 371-9088